

**Notice of Allowability**

Application No.

09/649,270

Examiner

Tuan A. Vu

Applicant(s)

CROWL ET AL.

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 2/17/2006.
2. ☒ The allowed claim(s) is/are 1,4-7,9-10,12-16 and 19-22 ( renum 1-16).
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some\* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\* Certified copies not received: \_\_\_\_\_.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.  
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.
- (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review ( PTO-948) attached
- 1) ☐ hereto or 2) ☐ to Paper No./Mail Date \_\_\_\_\_.
- (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date \_\_\_\_\_.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

- |   |   |
|---|---|
| 1. <input type="checkbox"/> Notice of References Cited (PTO-892)  | 5. <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)                       |
| 2. <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                | 6. <input checked="" type="checkbox"/> Interview Summary (PTO-413),<br>Paper No./Mail Date _____. |
| 3. <input type="checkbox"/> Information Disclosure Statements (PTO-1449 or PTO/SB/08),<br>Paper No./Mail Date _____ | 7. <input checked="" type="checkbox"/> Examiner's Amendment/Comment                               |
| 4. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit<br>of Biological Material          | 8. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance              |
|   | 9. <input type="checkbox"/> Other _____.  |

### DETAILED ACTION

1. This action is responsive to the Applicant's response filed 2/17/2006.

As indicated in Applicant's response, claims 1, 10, 16, 21 have been amended. Claims 1, 4-7, 9-10, 12-16, 19-22 are pending in the office action.

### EXAMINER'S AMENDMENT

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Fred Mollborn, Reg. 48, 5877 on 3/31/2006.

The application has been amended as follows.

In the Claims:

#### **Claim 1:**

A method of compiling a computer source program into a compressed-object code file containing differential names, said method comprising:

receiving a source program written in a high-level programming language, the source program including one or more program symbols and non-program symbol information, wherein at least some of the one or more program symbols form containing scopes for other program symbols, the containing scopes being selected from the group consisting of: a namespace, a package, a module, a container object and a function;

encoding each of the a program symbol name of the source program to produce an encoded program symbol name, without changing the non-program symbol information;

when upon determining that the encoded program symbol has a containing scope specified in the source program, determining a differential name for the encoded program

Art Unit: 2193

symbol name relative to a base symbol identifying a the containing scope for the program symbol as specified in the source program, wherein the containing scope defines a context within which the differential name is distinct with respect to the base symbol, and wherein the differential name contains fewer characters than the encoded program symbol name and is formed at least in part by a sequence of characters constituting a subset of the encoded program symbol name; and

replacing the encoded program symbol name with the differential name to facilitate producing a compressed object code file including the differential name.

**Claim 10:**

A method of generating encoded program symbol names in an uncompressed form from compiled information containing differential names, said method comprising:

receiving a source program including one or more encoded program symbol names, wherein at least some of the one or more program symbols form containing scopes for other program symbols, the containing scopes being selected from the group consisting of: a namespace, a package, a module, a container object and a function;

determining whether any encoded program symbol names are in a differential format;

for each program symbol name that is in a differential format:

extracting a differential program symbol name and a reference to a base symbol identifying a containing scope for the program symbol as specified in the source program, wherein the containing scope ~~is selected from a group consisting of: a namespace, a package, a module, a container object, and a function,~~ and defines a context within which the differential name is distinct with respect to the base symbol, and wherein the differential name contains fewer characters than the encoded program symbol name and is formed at least in part by a sequence of characters constituting a subset of the encoded program symbol name;

using the extracted reference to locate a non-differential name for the base symbol; and

replacing the differential program symbol name with the non-differential name for the base symbol containing scope to obtain an encoded program symbol name in an uncompressed form, the encoded program symbol name containing more characters than the differential name and including the differential name, whereby a decompressed compiler product including the encoded program symbol name is produced.

**Claim 13:**

A compiler system suitable for compilation and utilization of source programs, said ~~compilation~~ compiler system comprising:

an enhanced compiler suitable for generation of enhanced compiler products, the enhanced compiler being operable to: ~~compile a source program having at least one program symbol name to produce the enhanced compiler products, the enhanced compiler products having a reduced size in comparison to compiler products produced by conventional compilers and including one or more differential names corresponding to the program symbol names; and~~

receive a source program written in a high-level programming language, the source program including one or more program symbols and non-program symbol information, wherein at least some of the one or more program symbols form containing scopes for other program symbols, the containing scopes being selected from the group consisting of: a namespace, a package, a module, a container object and a function;

encode each of the program symbol name of the source program to produce an encoded program symbol name, without changing the non-program symbol information;

upon determining that the encoded program symbol has a containing scope specified in the source program, determine a differential name for the encoded program symbol name relative to a base symbol identifying the containing scope for the program symbol as specified in the source program, wherein the containing scope defines a context within which the differential name is distinct with respect to the base symbol, and wherein the differential name contains fewer characters than the encoded program symbol name and is formed at least in part by a sequence of characters constituting a subset of the encoded program symbol name; and

replace the encoded program symbol name with the differential name to facilitate producing a compressed object code file including the differential name;  
at least one enhanced non-compiler component operable to understand and utilize the enhanced compiler products; and  
a computing device operable to execute one or more of the enhanced compiler and the at least one enhanced non-compiler component.

**Claim 16:**

A computer readable medium including computer program code for compiling a computer source program into a compressed object code file containing differential names, said computer readable medium comprising:

computer program code for receiving a source program written in a high-level programming language, the source program including one or more program symbols and non-program symbol information, wherein at least some of the one or more program symbols form containing scopes for other program symbols, the containing scopes being selected from the group consisting of: a namespace, a package, a module, a container object and a function;

computer program code for encoding each of the a program symbol name of the source program to produce an encoded program symbol name, without changing the non-program symbol information;

computer program code for determining, ~~when~~ upon determining that the encoded program symbol has a containing scope specified in the source program, a differential name for the encoded program symbol name relative to a base symbol identifying a the containing scope for the program symbol as specified in the source program, wherein the containing scope defines a context within which the differential name is distinct with respect to the base symbol, and wherein the differential name contains fewer characters than the encoded program symbol name and is formed at least in part by a sequence of characters constituting a subset of the encoded program symbol name; and

computer program code for replacing the encoded program symbol name with the differential name to facilitate producing a compressed object code file including the differential name.

**Claim 21:**

A computer readable medium including computer program code generating encoded program symbol names in an uncompressed form from compiled information containing differential names, the encoded program symbol names being associated with compiler information, said computer readable medium comprising:

computer program code for receiving a source program including one or more encoded program symbol names, wherein at least some of the one or more program symbols form containing scopes for other program symbols, the containing scopes being selected from the group consisting of: a namespace, a package, a module, a container object and a function;

computer program code for determining whether any encoded program symbol names are in a differential format;

for each encoded program symbol name that is in a differential format:

computer program code for extracting a differential program symbol name and a reference to a base symbol identifying a containing scope for the program symbol as specified in the source program, wherein the containing scope ~~is selected from a group consisting of: a namespace, a package, a module, a container object, and a function,~~ and defines a context within which the differential name is distinct with respect to the base symbol, and wherein the differential name contains fewer characters than the encoded program symbol name and is formed at least in part by a sequence of characters constituting a subset of the encoded program symbol name;

computer program code for using the extracted reference to locate a non-differential name for the base symbol; and

computer program code for replacing the differential program symbol name with the non-differential name for the base symbol to obtain an encoded program symbol name in an

uncompressed form, whereby a decompressed compiler product including the encoded program symbol name is produced.

***EXAMINER'S STATEMENT OF REASONS FOR ALLOWANCE***

3. Claims 1, 4-7, 9-10, 12-16, 19-22 are allowed.

The following is an examiner's statement of reasons for allowance:

A method or product for producing compressed object code or compiler products, comprising (i) encoding each of the program symbol name of a received high-level programming source language; and upon determining that the encoded symbol has a containing scope, determining a differential name for the encoded program symbol name relative to a base symbol identifying the containing scope for the program symbol as specified by the source program; (ii) wherein the containing scope is selected from the a group consisted of a namespace, a package, a module, a container object and a function, and defines a context within which the differential name is distinct from the base symbol and contains fewer characters than the encoded program symbol name; and replacing the encoded program symbol name with the differential name to produce the compressed object code or compiler products; as recited in claims 1, 13 and 16.

A method or product for producing encoded program symbol names in uncompressed form, comprising (iii) receiving a source program comprising encoded program symbol names; and for each encoded program symbol name that is in a differential format, extracting the differential name for the encoded program symbol name relative to a base symbol identifying the containing scope for the program symbol as specified by the source program; (iv) wherein the containing scope is selected from the a group consisted of a namespace, a package, a module, a container object and a function, and defines a context within which the differential name is distinct from the base symbol and contains fewer characters than the encoded program symbol

Art Unit: 2193

name; and replacing the differential name with the non-differential name to obtain the uncompressed form of the program symbol names; as recited in claim 10 and 21.

The prior art taken separately or jointly does not suggest or teach the following features.

**Unger**, USPN: 5,991,713, for encoding HTML source file with encoded tag formatting using conventional compressing techniques to compress hypertext content, does not disclose a containing scope of a encoded program symbol name of a high-level program source code, such that the containing scope is specified from the source program as being a namespace, a module or a function as recited in (ii) or (iv) and identifying a base symbol relative to which the differential name is formed from encoded high-level programming source program symbols as recited in (i) and (iii).

**Storer**, “the Macro Model for Data Compression”, teaches encoding dictionary string of alpha characters and replacing common container with pointer structures; but does not suggest the container of encoded program symbols based on the approach of (i) and (ii).

**Ainon**, “Storing Text using Integer codes”, teaches similar methodology as Storer so that common family of alphabetical terms are grouped into special coded structures using common compression scheme, but does not teach or suggest the separation of encoded form of container and differential program symbol name as in (i) and (ii) from above.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

### ***Conclusion***

Art Unit: 2193

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence - please consult Examiner before using) or 571-273-8300 ( for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT  
March 31, 2005

  
KAKALI CHAKI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100